RECONSTRUCTION OF PERIODIC SONAR SIGNALS HIDDEN IN WIDEBAND NOISE USING ENSEMBLE AVERAGING AND MULTI-RATE DSP

Midshipman Samuel P.A. Peterson, Class of 2003
Professor Antal A. Sarkady
Commander Charles B. Cameron, USN

United States Naval Academy
Electrical and Computer Engineering Department
P.O. Box 14181
Annapolis, Maryland 21412

Abstract: The reconstruction of periodic acoustical signals with time domain periodic averaging requires a reliable estimate of the fundamental frequency (f_1) of the signal. The reconstruction task is particularly difficult when the signal is "hidden" in additive noise and the signal-to-noise ratio is poor. This is usually the case in most passive SONAR problems when early detection and characterization of targets is required. Statistically reliable estimates of the fundamental frequency of a noisy periodic signal can be computed in the frequency domain using Bartlett's smoothing procedure. In this procedure, a long, noisy signal is segmented into M mutually exclusive time segments and a power spectral estimate for each segment is computed. Spectral estimates are ensemble-averaged to enhance the signal power and reduce the residual spectral variance of the additive noise. In Bartlett's smoothing procedure the spectral line detection efficiency improves with $\sqrt{M}$ when $M > 50$.

The Bartlett's smoothing procedure merely provides a range of values for the fundamental frequency within a range of four times the standard deviation of the embedded periodic signal. In the reconstruction phase, the recorded noisy signal is reused to obtain one or more cycles of the "clean" signal. In the reconstruction procedure, the noisy signal is segmented into J mutually exclusive time segments, each exactly T seconds in length. Ensemble averaging in the time domain of these segments recovers the required "clean" signal with an enhancement efficiency of $\sqrt{J}$ when $N > 50$ and when the proper value of T is used. Because in most problems the correct value of T is not known, the enhancement procedure is iterated over a range of four times the standard deviation and that iteration which provides the maximum signal-to-noise ratio is declared the winner. For proper enhancement, an integer number of sample points must occur in T, for each choice of T. This requires a new sampling rate be used on the original time sequence for each choice of T. The resampling is efficiently achieved using an FFT interpolation technique. The algorithms are optimized for the SHARC ADSP-21060 DSP hardware and can be used in real time applications.

Key Words: Bartlett's smoothing; downsampling; DSP program; MEX; PC program; sister points; time domain periodic averaging; virtual resampling.

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE **APR 2003**	2. REPORT TYPE	3. DATES COVERED **00-00-2003 to 00-00-2003**

4. TITLE AND SUBTITLE **Reconstruction of Periodic Sonar Signals Hidden in Wideband Noise Using Ensemble Averaging and Multi-Rate DSP**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **United States Naval Academy,Department of Electrical and Computer Engineering,Annapolis,MD,21402**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S) 11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Proceedings of the 57th Meeting of the Society for Machinery Failure Prevention Technology, Virginia Beach, VA 173?185 (April 2003)

14. ABSTRACT

The reconstruction of periodic acoustical signals with time domain periodic averaging requires a reliable estimate of the fundamental frequency (fl) of the signal. The reconstruction task is particularly difficult when the signal is ?hidden? in additive noise and the signal-to-noise ratio is poor. This is usually the case in most passive SONAR problems when early detection and characterization of targets is required. Statistically reliable estimates of the fundamental frequency of a noisy periodic signal can be computed in the frequency domain using Bartlett?s smoothing procedure. In this procedure, a long, noisy signal is segmented into M mutually exclusive time segments and a power spectral estimate for each segment is computed. Spectral estimates are ensemble-averaged to enhance the signal power and reduce the residual spectral variance of the additive noise. In Bartlett?s smoothing procedure the spectral line detection efficiency improves with M when M > 50. The Bartlett?s smoothing procedure merely provides a range of values for the fundamental frequency within a range of four times the standard deviation of the embedded periodic signal. In the reconstruction phase, the recorded noisy signal is reused to obtain one or more cycles of the ?clean? signal. In the reconstruction procedure, the noisy signal is segmented into J mutually exclusive time segments, each exactly T seconds in length. Ensemble averaging in the time domain of these segments recovers the required ?clean? signal with an enhancement efficiency of J when N >50 and when the proper value of T is used. Because in most problems the correct value of T is not known, the enhancement procedure is iterated over a range of four times the standard deviation and that iteration which provides the maximum signal-to-noise ratio is declared the winner. For proper enhancement, an integer number of sample points must occur in T, for each choice of T. This requires a new sampling rate be used on the original time sequence for each choice of T. The resampling is efficiently achieved using an FFT interpolation technique. The algorithms are optimized for the SHARC ADSP- 21060 DSP hardware and can be used in real time applications.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**13**	

Introduction: In order for a submarine to detect another submarine submerged beneath the ocean, it must ordinarily clearly identify the presence of a distinctive periodic acoustic signal in the water, originating from the enemy submarine. The ocean, however, is an extremely noisy environment. Ambient acoustic noise from animal life, ships, weather, and other sources can drown out the oftentimes-faint periodic acoustic signatures of submarines. These periodic signatures can come from machinery in the target submarine rotating or operating in some other periodic fashion. For example, the shaft which turns the screws of the vessel in order to propel it through the water may provide such an acoustic signature.

Target Detection with Bartlett's Smoothing: The noisy acoustical voltage or current waveform, $v(t)$, observed at a remote listening sight can always be modeled in the time domain as the superposition of two events:

$$v(t) = s(t) + n(t) \tag{1}$$

where $s(t)$ is the signal arriving from a distant source and $n(t)$ is the ambient ocean noise. When these two events are mutually orthogonal (i.e. the cross-correlation [11, 10, 1] of $s(t)$ and $n(t)$ is zero), then the power spectrum of the noisy waveform, $P_v(\omega)$, is:

$$P_v(\omega) = P_s(\omega) + P_n(\omega) \tag{2}$$

where $P_s(\omega)$ and $P_n(\omega)$ are the power spectra of the clean signal and the noise, respectively [8, 10]. In most sonar problems, the assumption of orthogonality has been found to be valid [10]. A similar statement can be made for the magnitude spectra of these events.

$$A_v(\omega) = A_s(\omega) + A_n(\omega) \tag{3}$$

The detection of a distant target must be based on statistically reliable spectral estimates. A reliable estimate is one that is not unduly subject to random variations. The statistical reliability of spectral estimates (power or magnitude) can be improved by a procedure called Bartlett's smoothing. The signal enhancement is assured by the central limit theorem [9, 8, 10] and provides improvement in the signal-to-noise ratio.

The mathematical algorithm known as Bartlett's smoothing (see fig. 4) [8] is used to average out the noise component of the waveform so that only the signal is left. In this procedure, a long, noisy signal is segmented into *M* mutually exclusive time segments and a magnitude spectral estimate for each segment is computed. A typical segment of unpadded, noisy data made up of a 0.1 V peak, 100 Hz sine wave hidden in 1.0 V RMS noise with even distribution is shown in figure 1. Each sample data segment (element of the time domain ensemble) is zero padded in the time domain (at the end), extending its length from 1,028 samples to 16,384 samples before performing the

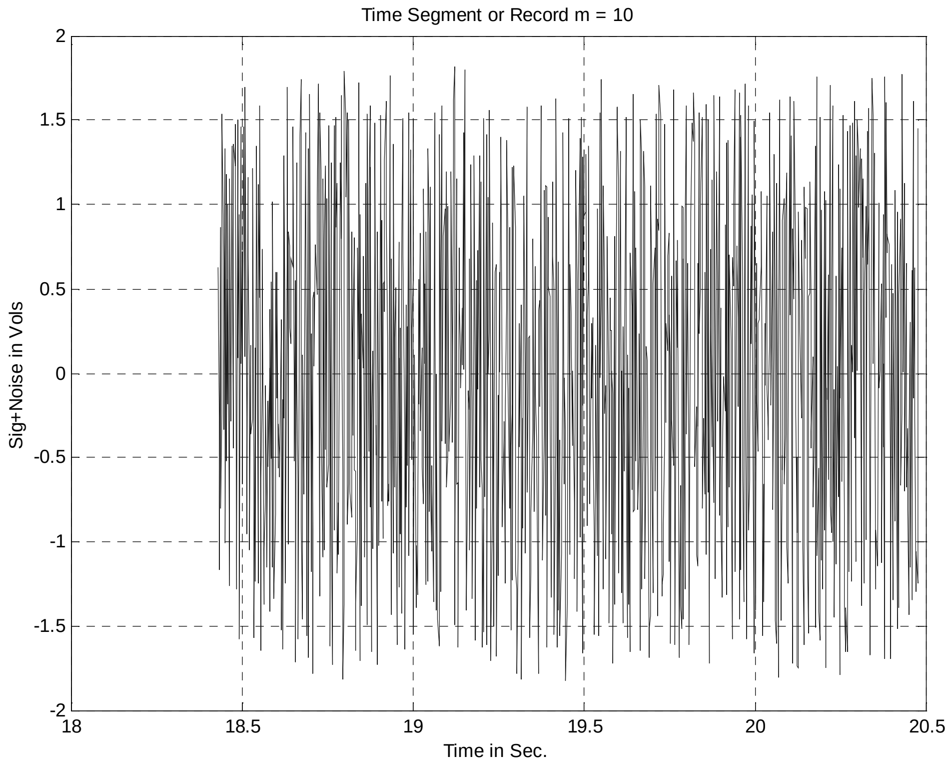
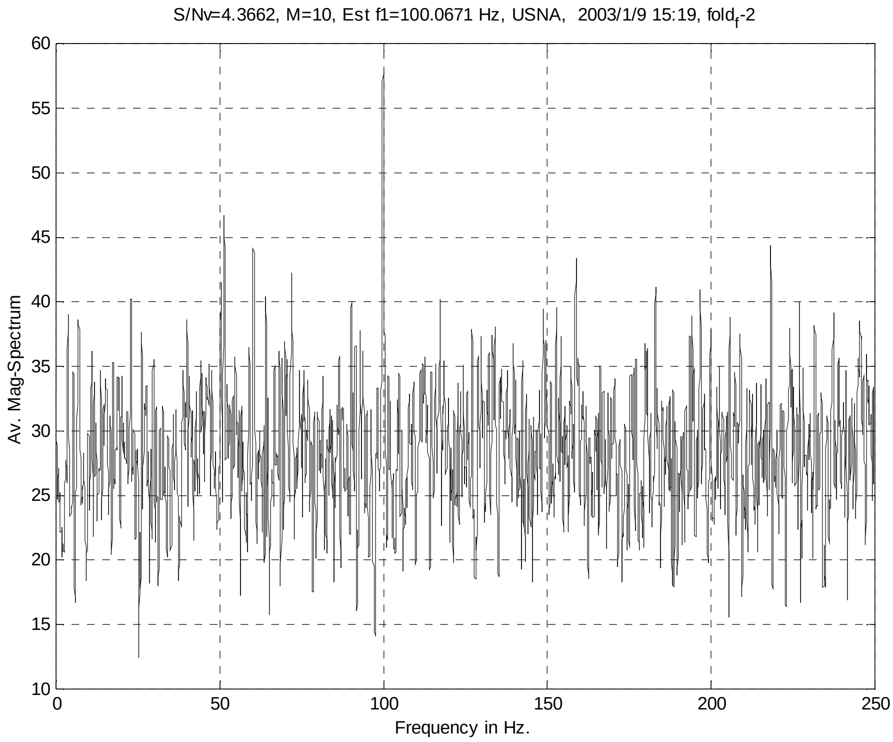


Figure 1: 0.1 V_{peak}, 100 Hz sinusoid hidden in 1.0 V_{rms} noise.

complex, radix-2 fast Fourier transform (FFT). Zero padding in the time domain (increasing the data length by a factor of 16), reduces the picket-fence effect [2] in the frequency domain and improves the frequency resolution by a factor of 16 [1, 3, 4, 5]. In Bartlett's smoothing procedure, ensemble averaging produces spectral line enhancement, while reducing spectral variance due to additive noise. In this procedure, spectral-line detection efficiency improves as $\sqrt{M}$ when $M > 50$ (where M is the number of elements in the ensemble). A spectral-estimate based on $M = 10$ is shown in figure 2. Note that the 100 Hz spectral line just barely emerges from the noise with a signal-to-noise voltage ratio (S/Nv) of 4.3662. In taking this ratio the signal is expressed in peak voltage, whereas the noise is expressed in RMS voltage. Compare this to the spectral-estimate based on a $M = 40$ (shown in fig. 3) where the signal-to-noise ratio is 6.4896 and the 100 Hz spectral line more clearly rises out of the noise floor. A higher signal-to-noise ratio yields a greater target detection confidence. The S/Nv of 6.4896 gives approximately 98% detection confidence and only a 2% chance that the spectral line is due to a statistical anomaly [8, 6, 7].



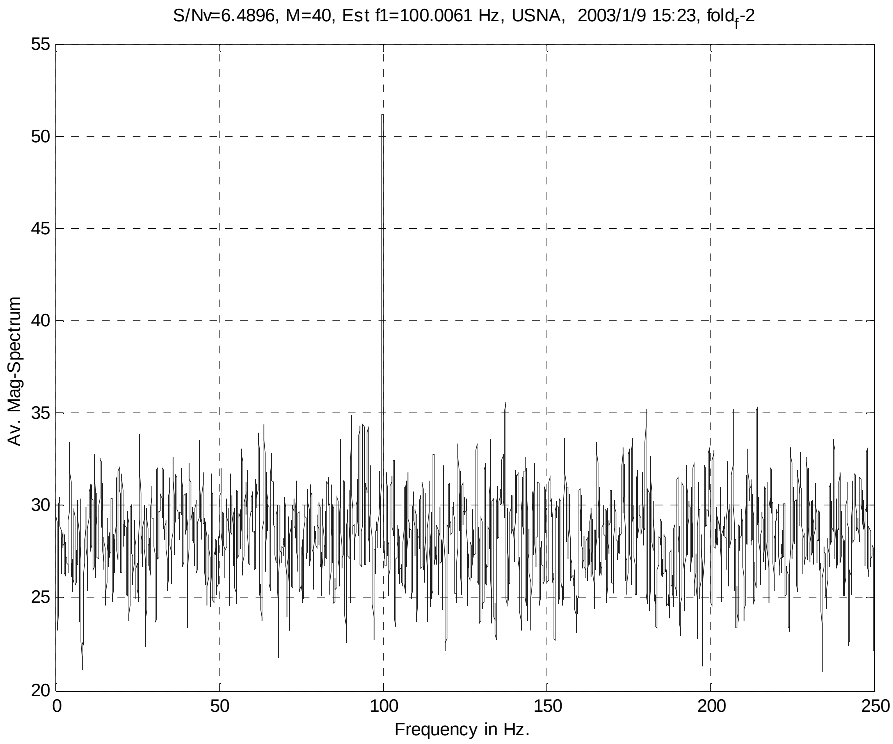


Figure 2 (top): Results of Bartlett's smoothing, $M = 10$, S/Nv = 4.3662
Figure 3 (bottom): Results of Bartlett's smoothing, $M = 40$, S/Nv = 6.4896

The exact frequency of the signal must be known in order to identify the source of the signal. The question now arises: how can the frequency analysis be refined in order to pinpoint the signal's frequency with a sufficient degree of accuracy? There is another frequency analysis process called time domain periodic averaging which can test a waveform for the presence of a periodic signal with a specific fundamental frequency. What, then, was the utility of Bartlett's smoothing? Why did we not just perform time domain periodic averaging in the first place? The answer is that periodic time averaging only tests for one specific frequency, revealing the strength of that frequency alone. The overall frequency range in which a contact's frequency might fall is too large to simply test all possible frequencies. That would be inefficient and would consume more processing time than can be afforded for the proposed real-time application. Instead, Bartlett's smoothing procedure is used as a first and continuously running method of detection of a contact, as well as a way to narrow the range of possible frequencies to a practical size once that contact is detected.

A good analogy for the interaction of Bartlett Smoothing and time domain periodic averaging is that of a lookout on a ship. That lookout has two tools available to him for detecting and identifying a contact at sea: his naked eyes and his binoculars. First, he uses his naked eyes, which cannot see a contact as clearly as with the binoculars, to scan the entire angular range for which he is responsible. He keeps scanning with the naked eye until he detects that a contact is present, at which time he uses his binoculars to scan only that small region where the contact was seen. His binoculars will allow him to examine a small area in detail and to accurately identify the contact. Bartlett's smoothing corresponds to the naked eye in this analogy, and time domain periodic averaging corresponds to the set of binoculars. If the lookout constantly scanned only small areas with his binoculars, he might not detect a contact until it was too late to avoid a collision or maybe until the contact left the visual identification area. Just as the lookout is constrained by the time it would take him to scan the entire horizon bit by bit with binoculars, we are constrained by processing speeds in the digital signal processor which are the centerpiece of this research.

Bartlett's smoothing will run continuously until it finds something conclusive. After averaging a specified number of segments together, the algorithm finds the maximum point of the ensemble. If the difference between the noise floor and the maximum is not at least thirty-six times the standard deviation of the noise floor of the ensemble, Bartlett's smoothing repeats the process, continuing to average new segments into the ensemble. If this test is eventually passed, the signal-to-noise ratio is great enough to determine with confidence that a periodic signal is present in the noise. The reason for the number thirty-six is that it is the square of six, the signal-to-noise ratio which indicates the presence of a target with greater than 90% confidence. We process the results of Bartlett's smoothing in terms of power (vice magnitude), because the square root operation required to compute magnitude is too costly in terms of processing time (see fig. 4.) The orthogonality of signal and noise makes this possible (see Eqs. (2) and (3).) Let the frequency corresponding to the maximum magnitude of the ensemble be $f_{\max}$. The question now becomes how to determine the frequency range around $f_{\max}$ to be tested by the time domain periodic averaging. Let the beginning and end of the frequency range to be tested by the time domain periodic averaging be f_1 and f_2 respectively (see fig. 4.)

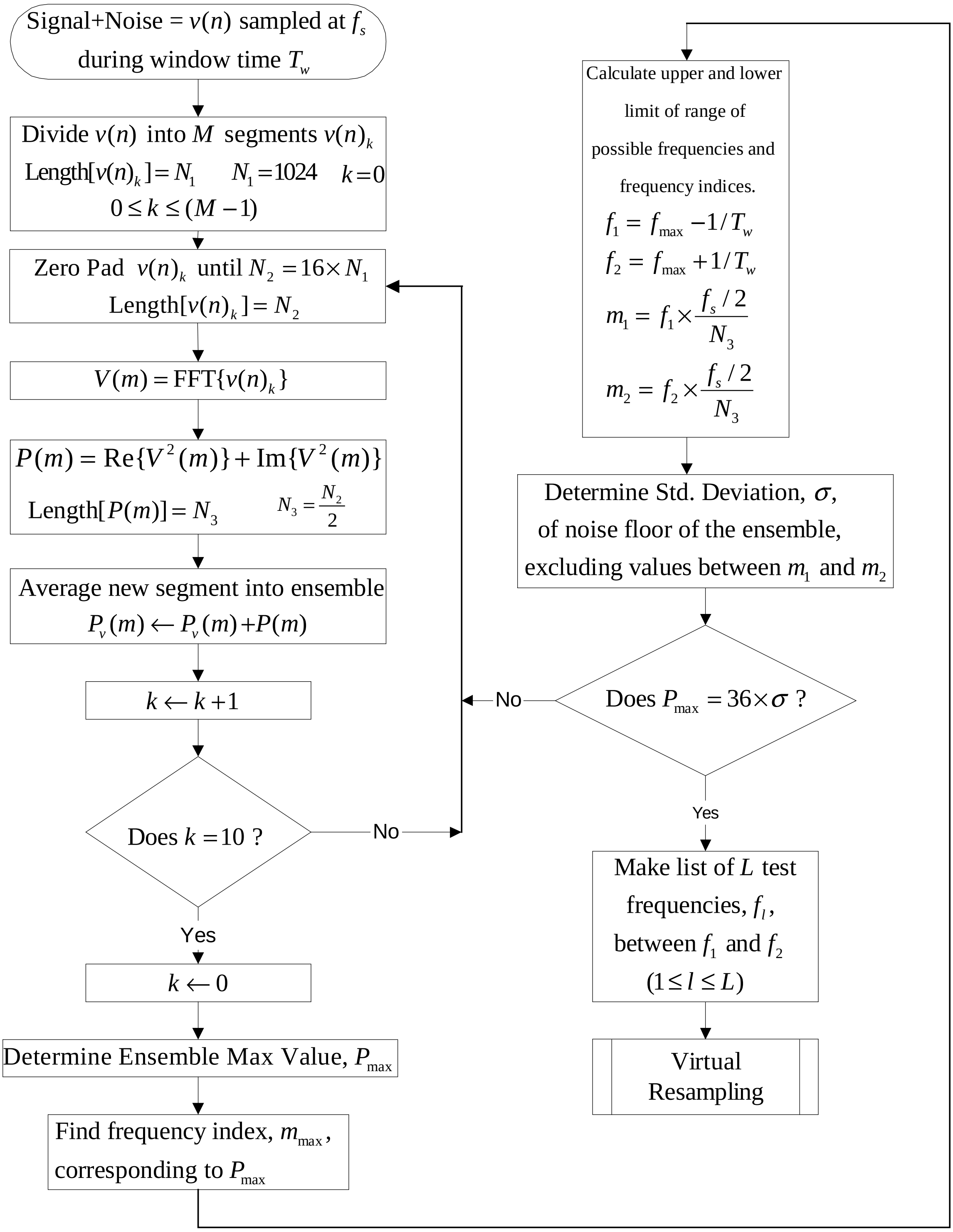


Figure 4. Bartlett's Smoothing Procedure

Suppose that after recording waveforms of acoustic noise in the ocean for a while, Bartlett's smoothing procedure tells us that there is a contact out there somewhere because a periodic signal within the range of 99-101 Hz has been detected. Now, we must take that waveform and perform the periodic time averaging operation on it for specific frequencies between 99 and 101 Hz, for example 99.0, 99.5, 100.0, 100.5, and 101.0 Hz. The periodic time averaging procedure allows us to test the waveform for the strength of one particular frequency, so we must perform it over and over for each distinct frequency we wish to test for.

Target Identification with Time Domain Periodic Averaging: Now that a target has been detected with a high degree of confidence, we can proceed with the target characterization and identification phase. Our method of target characterization is to reconstruct one cycle of the periodic signal using time domain periodic averaging. This process is performed using the same original time series data already used in the Bartlett's smoothing procedure. The reconstructed cycle will be used as a template for target identification. In order to reconstruct the signal, its fundamental frequency must be known beforehand; this information is obtained from Bartlett's smoothing procedure. The first step in the time domain periodic averaging procedure [9, 10, 11] (see fig. 6) is to take one of the frequencies, f_l, from the list of possible frequencies provided by Bartlett's smoothing and then segment the original waveform into segments with length equal to the corresponding period, $T_l = 1/f_l$. The corresponding time index and periodic index are n and N_5, respectively (see fig.6.) The original waveform is then

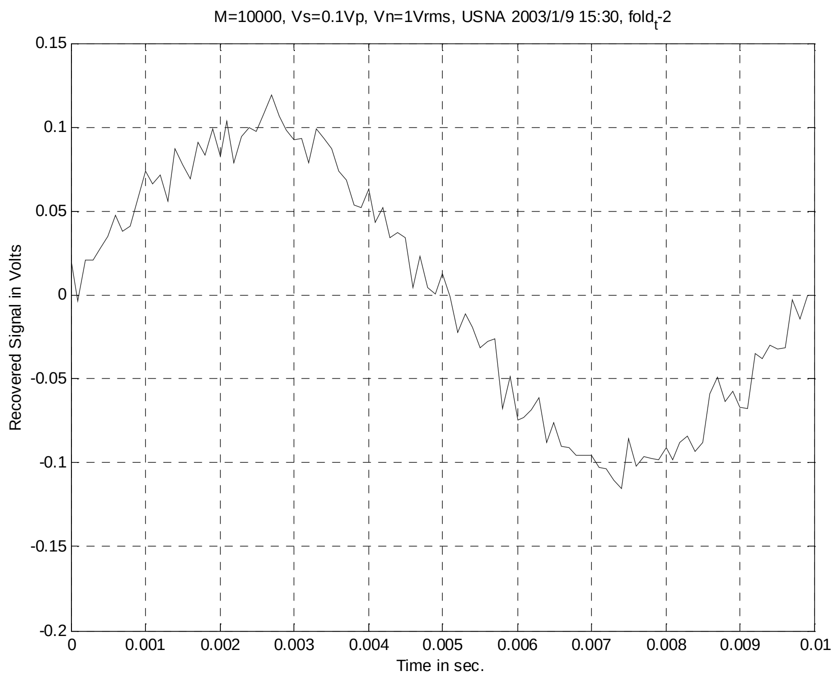


Figure 5: Results of Time Domain Periodic Averaging

segmented (in the time domain) into equal segments of length N_5. At this point, the averaging begins. We average the value of the waveform at time index n, which is within the first segment of data, with the values at time indices $n+N_5$, $n+2N_5$, $n+3N_5$, ...and $n+(J-1)N_5$, where J is the number of segments into which the waveform was originally divided. Thus we take the value at a point in the first segment and average it

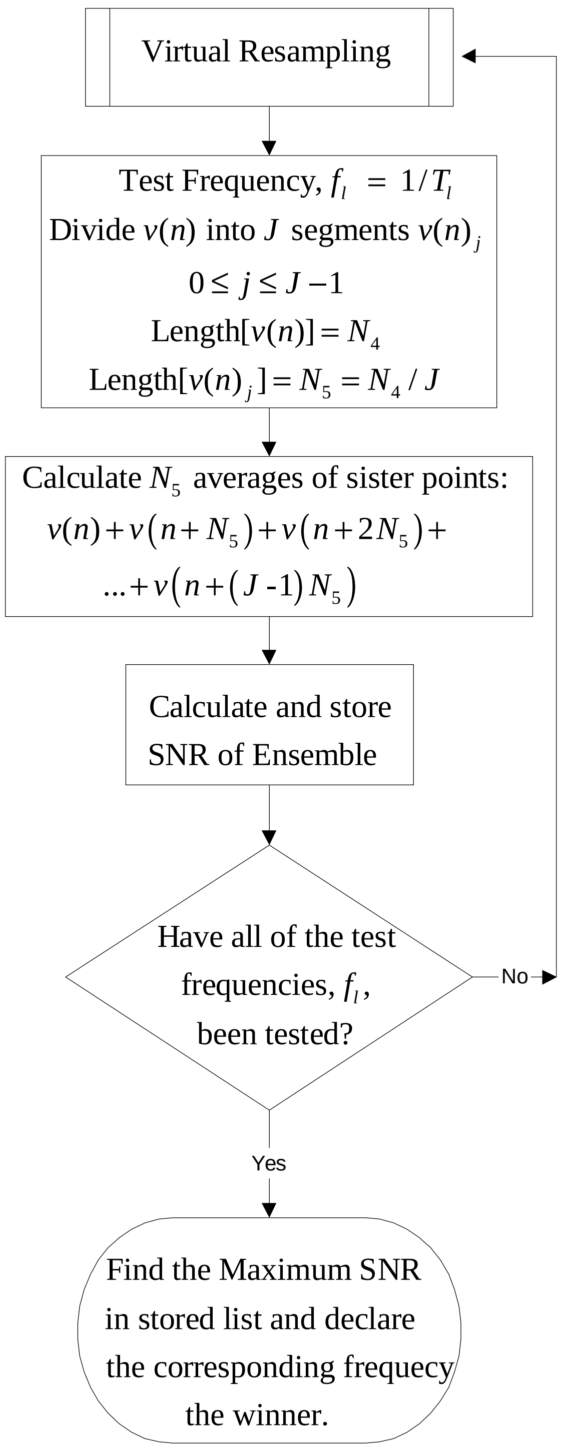


Figure 6: Time Domain Periodic Averaging

with all the values at its sister points in each of the other segments. (Sister points are points separated by an integral number of periods.) This operation is performed for each point within the segment. The result of this will be an ensemble composed of the average of the segments (see fig. 5).

Like Bartlett's smoothing, this process takes advantage of the fact that noise is random, and therefore the average of the noise will be close to zero. The average of the periodic signal within the waveform will become firm in its value when averaged with its sister points on other cycles of the signal. This is how the periodic time averaging improves the signal-to-noise ratio. It is evident that the whole procedure begins with and hinges upon the selection of the frequency to be tested, since that is what determines the period which will constitute the segment length and which will cause the signal-to-noise ratio to increase when averaging when averaging the sister points of the. A very exact estimate of the signal frequency is required in order to yield useful results. There is another extremely important, practical issue concerning implementation of these algorithms which has not been considered yet.

Since a digital signal processor will be used to perform the mathematical algorithms which have been described, the waveform must be discrete, that is to say that the continuous waveform recorded by the hydrophones must be sampled by an analog to digital converter in order to be processed by the digital signal processor. This means we will run into a problem when we try to perform time domain periodic averaging. The time domain periodic averaging was explained as if a continuous waveform was being processed. A discrete waveform, however, only has value at particular, evenly spaced points, and has no value whatsoever between those sample points. The distance between points (the sampling period) in the sampled waveform is $T_s = 1/f_s$, where f_s is the sampling frequency used to capture the waveform. It will be impossible to average the sister points of the different segments unless T_t (the segment length) is an integral multiple of T_s. T_t is determined by f_t, the frequency for which we wish to test, so we must engineer T_s to divide T_t evenly, or in other words, $T_t \bmod (T_s) = 0$. This presents another problem. Waveforms are sampled by the ADC in real time and all continuous data in the waveform are forever lost. It is impossible to physically resample the original waveform, since that waveform represents a sound wave that existed only at the particular moment in time that it was sensed by the hydrophones. One possible solution to this problem would be to use analog recording techniques such as magnetic tape to record the waveform for resampling at a later time. This would mean the original waveform would need to be physically resampled in order to make T_s divide T_t evenly for each frequency to be tested using periodic time averaging. The amount of processor memory and increased time this would require rules it out as a solution.

Virtual Resampling: Since it is out of the question to physically resample the waveform for every run of the time domain periodic averaging, a method called virtual resampling is used. Virtual resampling is the process of effectively multiplying the sampling period,

T_s, by some factor (possibly non-integral) so that $T_t \bmod (aT_s) = 0$ where a is the multiplication factor and T_s is the original, physical sampling period with which the waveform was captured. This method consists of two steps: upsampling and downsampling. Upsampling multiplies T_s by an integral factor, n, and downsampling divides T_s by an integral factor, k. Therefore, $a = n/k$.

Upsampling is accomplished by zero padding in the frequency domain. The original, sampled waveform is converted to the frequency domain using the radix-2 fast Fourier transform. The resulting magnitude of the FFT is an array of N discrete values. The array length is multiplied by a factor of n by placing an array of $(n\text{-}1)\times N$ zeros in the middle of the array, separating the first half and second half. This discrete, frequency domain representation of the waveform is simply an array of values, except that after the zero padding the series of values is no longer associated with specific points on the frequency axis. To the DSP, the series of values in the frequency domain is just an array. The inverse FFT (IFFT) is then performed on this zero-padded array, and since the length of the array in the frequency domain is equal to the length of the array produced by the IFFT in the time domain, the number of sample points in the time domain has been multiplied by a factor of n as well. The length of the time domain signal in units of time, however, remains the same as that of the original waveform. So now there are n times the number of sample points per unit of time there were before upsampling. Practically speaking, the resolution of the waveform has been improved by a factor of n by reducing the amount of time between samples. What has been achieved is interpolation in the time domain. The only limit on how finely we can interpolate is the amount of memory available to us to store the array after we have enlarged it.

Downsampling is extremely simple. It consists of decimation of the upsampled waveform in the time domain by an appropriate factor, k. This means that the upsampled array is sampled at every k^{th} value. By upsampling an array and then downsampling the resultant array, the original sampling rate has been effectively multiplied by the rational factor n/k, thus virtually resampling the original waveform. As indicated earlier, this virtual resampling must be done for every possible, specific frequency that is to be tested from the narrow range of possible frequencies given by the results of Bartlett's smoothing.

Implementation: The FFT code used in our algorithm was written in assembly language by Analog Devices, the maker of the ADSP-21060 SHARC DSP, and is freely available on their website [15]. It is a most efficient, complex, radix-2 FFT implementation on the SHARC DSP. This program was altered in only minor ways to make it callable from another program which also runs on the DSP and is written in the C programming language. This program generates the sine and cosine factors used in the calculation of the FFT and is under the control of another program which runs on the PC. The PC program and the DSP program interact through a two-way handshake scheme. The DSP program also pads the real and imaginary input arrays with zeros for increased resolution in the output (virtual resampling).

The PC program, also written in C, controls the operation of the DSP through the Blacktip PCI (rev. 3) board on which it is mounted. It uses a series of pre-defined functions provided by Bittware, the maker of the Blacktip PCI board. These functions allow the PC to write and read data to and from any register in the DSP's memory, as well as give the DSP many other commands. The PC program simply downloads the real and imaginary arguments of the FFT and the length of the FFT to the specific locations in memory where the DSP will look for them and then calls the DSP program which performs the FFT.

The PC program is itself called by a MATLAB m-file which uses a convenient feature of MATLAB called MEX to call a C program as if it were a MATLAB function. The MATLAB m-file passes the input arguments (real and imaginary input arrays and the length of the FFT) to the PC program [12, 13, 14].

The PC program was written in C using Microsoft Visual C++ 6.0, and the DSP program (written in both C and assembly language) was created using the Analog Devices VisualDSP Software Development Tools version 2.2. VisualDSP produces an executable program file that can be downloaded to and run by the DSP.

The greatest possible length of our FFT on the ADSP-21060 is 16,384 because of the size of the internal memory. A greater length could be achieved if external memory were utilized, but the extra time those external memory accesses would take is unacceptable for our purposes. The ADSP-21060 has 4 Mbits of internal memory into which the program code must be stored along with the real and imaginary input and output data arrays, and the arrays of sine and cosine factors (used for computing the FFT.) Each sinusoidal array is half the length of the FFT, so the total number of 16,384 element arrays that must be stored is effectively five. Each element of each array is a 32-bit number in IEEE single-precision, floating-point format. Multiplying $16384 \times 32 \times 5$ yields a total data storage of 2.62 Mbits. Since we are using the radix-2 FFT for the sake of speed, the FFT length can only be powers of two. It is apparent, then, that 16,384 is the greatest possible length of the FFT, because the next greatest length, 32,768, would require more internal memory than is available on the ADSP-21060, and as noted above, we elected to use only internal memory.

To obtain an accurate measurement of the time required for the DSP to compute a single FFT (see table I), the programs were configured to compute the same FFT 1,000 times. This was done without the MATLAB program, and the input data were read by the PC program from data files. A stopwatch was used to time the completion of these 1,000 FFT's with consistent results. Because of the significant amount of time required by the calculation of 1,000 FFT's, the human error induced by the use of the stopwatch was deemed insignificant. The stopwatch was started simultaneously with the keyboard command which started the calculation and was stopped by visual cue from the computer screen, the disappearance of the DOS window indicating that the PC program had finished working. Also, because of the large number of FFT's, the error from the other parts of the program besides the FFT itself was deemed to be within acceptable limits.

Time Trials for the FFT	
Length (N): 16,384	
Processor: ADSP-21060 SHARC DSP	
Number of FFT's: 1,000	
	Time/s
Trial 1	40.7
Trial 2	40.7
Trial 3	40.6
Trial 4	40.8
Trial 5	40.7
Average of Trials	40.7
Average Time per FFT	**0.0407**

Table I: Speed of the FFT on the SHARC DSP

Conclusion: The most challenging problem in the signal reconstruction algorithm is to perform it in real time without introducing gaps in the acquired data. The question is, can the signals be processed as fast as they are recorded? Thanks to the relatively recent advances in DSP technology, the answer is "yes." Modeling the ocean as a 4 kHz bandwidth limited acoustic channel requires data to be sampled at 8 kHz in order to avoid aliasing, according to Nyquist theorem [10]. Using this 8 kHz sampling rate, the time required to gather 1024 samples of acoustic data is 1,024/8,000 s, or 128 ms. Each set of 1,024 samples is zero-padded to a length of 16,384. Since each 16,384-point FFT takes approximately 41 ms of computation time, 87 ms are left each cycle to complete the remainder of the algorithm. This is ample, so the algorithm can indeed be performed in real time.

References

1. Boaz Porat, "A Course in Digital Signal Processing", 1st. Edition, John Wiley & Sons, Inc., 1997, **ISBN: 0-471-14961-6**.

2. G. D. Bergland, "A Guided Tour of Fast Fourier Transform", IEEE Spectrum July 1969, pages 41-52.

3. Alan V. Oppenheim and Ronald W. Schaler, "Digital Signal Processing", Prentice-Hall, Inc. 1975.

4. Lawrence R. Rabiner and Bernald Gold, "Theory and Applications of Digital Signal Processing", Prentice-Hall, Inc. 1975.

5. John G. Proakis, Dimitris G. Manolakis, "Digital Signal Processing, Principles, Algorithms and Applications", 3rd Edition, Prentice-Hall Inc., 1996, **ISBN: 0-13-373762-4**.

6. David F. Mix, "Random Signal Processing", Prentice-Hall Inc., 1995, **ISBN: 0-20-381852-2**.

7. Manson H. Hayes, “Statistical Digital Signal Processing and Modeling”, John Wiley & Sons Inc., 1996, **ISBN: 0-471-59431-8**

8. Jenkins and Watts, "Spectral Analysis and its Applications", Holden - Day, 1968.

9. Peyton Z. Peebles, Jr., “Probability, Random Variables, and Random Signal Principles", 3rd Ed., McGraw-Hill Book Co., 1993, **ISBN: 0-07-049273-5**.

10. Ferrel G. Stremler, "Introduction To Communication Systems”, 3rd. ed., Addison-Wesley Publishing Co., 1990, **ISBN: 0-201-18498-2**

11. Julius S. Bendat and Allan G. Piersol, "Engineering Application of Correlation and Spectral Analysis", John Wiley and Sons, 1980.

12. Virginia Stonick and Kevin Bradley, “Labs for Signals and Systems Using MATLAB”, PWS Publishing Company, 20 Park Plaza, Boston, MA 02216-4324, Copyright 1996, **ISBN: 0-534-93808-6.**

13. Duane Hanselman and Bruce Littlefield, “The Student Edition of MATLAB, Version 4”, 1995, Prentice -Hall Inc., **ISBN: 0-13-184979-4.**

14. Vinary K. Ingle, John G. Proakis, “Digital Signal Processing Using MATLAB V.4”, PWS Publishing Company, 20 Park Plaza, Boston, MA 02116-4324, Copyright 1997, **ISBN**: **0-534-93805-1.**

15. Steven Cox, et al., “fftrad2.asm,” http://www.analogdevices.com , accessed 26 October 2002.